SCC.441 Information System Security Management

"Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2"

—Individual Report—

Ben Goldsworthy, 32098584
MSc Cyber Security

April 28, 2018

1 Introduction

WPA2, the 14-year-old protocol that provides secure Wi-Fi connectivity, the security of which had been formally proven, has been kracked. It has been shown to be fundamentally insecure at a specification level. This translates to different degrees of impact at the implementation level, depending on decisions taken by different vendors. That this attack uses a novel mechanism also suggests that a review of other key-based security protocols may turn up further vulnerable examples. Perhaps most crucially, this attack raises important questions about the writing of standards and access to them. This report shall establish the necessary background knowledge, summarise the findings of Vanhoef & Piessens (2017) and analyse their likely impact in the real world.

2 Background

In the beginning, there was Wired Equivalent Privacy (WEP). As Vanhoef & Piessens (2017) relate in §1 of their paper, this Wi-Fi access protocol was broken first in theory by Fluhrer et al. (2001), and then in practice by Stubblefield et al. (2002). In response, the Institute of Electrical and Electronics Engineers (IEEE) made the Wireless Protected Access (WPA) protocol available in 2003 as a draft of their eventual 802.11i amendment to the 802.11 standard. This protocol's operation revolved around the use of a 4-way handshake, and offered the use of two data-confidentiality and integrity protocols: (WPA-)TKIP (henceforth TKIP) as standard, and (AES-)CCMP (henceforth CCMP) optionally. A year later, the final version of 802.11i was released and brought with it Wireless Protected Access II (WPA2). The major difference between the two editions of WPA was WPA2's reversed approach to the two data-confidentiality and integrity protocols: CCMP was mandatory, and TKIP now optional. This was just as well, as TKIP was only ever intended as a interim solution, backwards-compatible with WEP devices, and was similarly insecure (Vanhoef & Piessens 2013). Additionally, in 2012, the 802.11ad amendment added the Galois/Counter Mode Protocol (GCMP) as another data-confidentiality option. This is part of the Wireless Gigabit (WiGig) proposal, which Grand View Research (2016) estimate will approach a market size over $7bn by 2024.

For 14 years, WPA2 has remained unbroken. Vanhoef & Piessens point out that whilst "[...]several attacks against protected Wi-Fi neworks were discovered over the years, these did not exploit flaws in 802.11i". All successful attacks targeted other aspects of the setup, such as the Wi-Fi Protected Setup (WPS), flawed random number generation, etc. Indeed, He et al. (2005) formally proved both the 4-way handshake and CCMP as secure. It is directly from this self-congratulatory atmosphere of complacency that this new form of attack has emerged, which may help to understand why it has come as such a shock to the industry.

3 Vanhoef & Piessens (2017)

3.1 How the attack was discovered

Vanhoef (2017) describes his original question, prompted whilst "slacking off, because I was supposed to be just finishing [another] paper [on OpenBSD's implmenation of the 4-way handshake]", as being about the `ic_set_key()` function, which installs the pairwise key (see below). "I wonder what happens if that function is called twice", he writes. The author's original guess—that it "might reset the nonces associated to the key"—proved to be correct for OpenBSD and potentially, he thought, for other vendors too. After finishing the paper, the author revisited his discovery and found that the flaw was not specifically in OpenBSD's implementation of the 4-way handshake, but in the specification of the 4-way handshake itself.

3.2 How the key reinstallation attack works

The 4-way handshake takes place after an initial authentication and (re)association stage when a client—or 'supplicant'—(re)connects to an access point (AP)—or 'authenticator'. It works as follows:

1. The authenticator sends the supplicant a message containing a nonce
2. The supplicant derives a Pairwise Transient Key (PTK) from this nonce and its own, and then sends its own nonce back to the authenticator
3. With the supplicant's nonce, the authenticator can also derive the PTK. It then sends the Group Temporal Key (GTK) back to the supplicant with an incremented replay counter
4. The supplicant replies to say it has received the GTK, then installs both it and the PTK. The authenticator, on receipt of this message, also installs the PTK.

Each of these four steps has an associated message. The issue arises from the specification's requirement that the supplicant accept retransmissions of message 3 from the authenticator, even after it has installed the PTK. This can be abused to "force a reinstallation of the PTK". This can be as simple as a channel-based man-in-the-middle attack that drops message 4 from the supplicant. After a set amount of time, the authenticator will resend message 3, which will prompt the supplicant to reinstall the key, resetting the replay counter and nonces whilst doing so, and thus allowing traffic to be decrypted. Variant implementations that only accept retransmitted message 3s immediately after original ones, or require the message 3 to be encrypted, are also dealt with.
Following this, the group key handshake is broken using the same technique. The not-widely-used PeerKey handshake is also breached, although the authors accept that "the result of [their] attack against the PeerKey handshake is rather low" due to its limited usage. Finally, the 802.11r Fast Basic Service Set (BSS)

Transition (FT) handshake is similarly laid low—the only example of the key reinstallation attack as applied to the AP, rather than the client.

3.3 The immediate implications

The immediate threat depends primarily on the data-confidentiality and integrity protocol used with WPA. If CCMP is used, an attacker is limited to replaying and decrypting packets. Though the authors mention Fouque et al. (2008) as discussing theoretical message forgery attacks against CCMP, they conclude that "[...]the attacks are theoretic and cannot be used to forge arbitrary messages." Victims using the (deprecated) TKIP are additionally vulnerable to arbitrary packet forgery, but only from the client to the AP. GCMP, however, fares far worse, suffering from all the previously-mentioned threats as well as arbitrary packet forgery in any direction. The impact is described as "[...]catastrophic[...]" and, when tied with the aforementioned predicted growth in WiGig, presents "[...]a worrying situation[...]". Additionally, the ability to forge packets from the client to the AP allows those packets to be forwarded on by the AP, allowing the vulnerability to be used as "[...]a gateway to inject packets towards *any* device connected to the network." In short, the confidentiality and integrity elements of the CIA triad are compromised in some way by this attack, whilst availability is untouched.

There are, however, a handful of caveats that mitigate the real-world risk somewhat. Thomson (2017) points out that "[...]an eavesdropper has to be in wireless range of the target network, and have the time and specialized software to pull off the KRACK technique." Whilst he points out that "[t]here is no, to the best of our knowledge, working exploit code available yet", Vanhoef claims to have "[...]made scripts to detect whether a [handshake implementation] is vulnerable to key reinstallation attacks [and that] these scripts will be released once we have had the time to clean up their usage instructions." He also promises to release "[...]a proof-of-concept script that exploits the all-zero key (re)installation present in certain Android and Linux devices[...]once everyone has had a reasonable chance to update their devices[...]".

More importantly, as Arnold KL tells Leyden (2017), "[...]a significant amount of the risk would be mitigated for services that use strong encryption at the transport or application layer (such as TLS, HTTPS, SSH, PGP) as well as applications secured by encrypted VPN protocols." He adds that this does not, however, completely mitigate the risk of metadata leakage.

3.4 The broader implications

The broader research questions to be answered by Vanhoef & Piessens thus became:

1. If the flaw is inherent in the standard, how many implementations are affected?

2. How can this flaw be compatible with the formal verification of the standard?

3. How applicable is the key reinstallation attack to alternative handshake methods?

To answer the first question, the attacks were performed against a range of different implementations—from iOS to Windows, Android to OpenBSD. This uncovered two points of particular interest. iOS and Windows were found to be protected against one of the main threats—the 4-way handshake attack—due to misimplementation of the standard. On the flipside of the coin, certain versions of Linux and Android (those that use `wpa-supplicant` 2.4 or higher) were extraordinarily vulnerable due a bug that provided an all-zero encryption key upon key reinstallation. This is attributed to misinterpreation of a vague section of the 802.11 standard, and the authors suggest that "[...]31.2 % of Android smartphones are likely vulnerable to the all-zero encryption key vulnerability."
On top of this, Leyden (2017) reports that "[i]t affects WPA2 Personal and Enterprise, regardless of the encryption ciphers used by a network" and once again points out that great Achilles heel of the brave new IoT world in that whilst "[c]omputers and modern phones are easy to patch,[...]it's the embedded devices and Internet-of-Things gizmos that are tricky to upgrade to address the WPA2 flaw."
Vanhoef & Piessens go on to examine how He et al. (2005) failed to catch this vulnerability in their formal analysis of the 4-way and group key handshakes. They conclude that He et al. "[...]proved that the 4-way handshake provides key secrey and session authentication[...]" and "[...]key ordering and key secrecy for the group key handshake[...]", none of which are violated by this attack, but that they "[...]do not model key installation[...]".
Finally, since publication of the paper, Vanhoef reports the discovery "[...]that the TDLS handshake and WNM Sleep Mode Response frame are also vulnerable to key reinstallation attacks." As this is a wholly novel type of attack, there is a strong potential for that list to grow as time goes on.

3.5 What can be done about it

Thankfully, the countermeasures given by Vanhoef & Piessens are remarkably simple: "First, the client implementing the data-confidentiality protocol should check whether an already-in-use key is being installed [and, i]f so, it should not reset assocated nonces and replay counters." Alternatively, the client implementing the data-confidentiality protocol should only install the a particular key once. These are easily implemented, and as such expect the US-CERT list of vulnerable devices to shrink quite rapidly.[1]
Moreover, modern cyber security approaches emphasises that one should never trust the channel, and this is a prime example of why not. As previously mentioned by Leyden, use of other secure protocols such as HTTPS and TLS, or a

[1] `https://www.kb.cert.org/vuls/byvendor?searchview&Query=FIELD+Reference=228519&SearchOrder=4`

VPN, will largely mitigate the risk of this attack.

4 Conclusion and further considerations

Vanhoef & Piessens have demonstrated an entirely new form of attack, and one that finally cracks a protocol assumed for 14 years to be safe. In doing so, they present a strong example of the threat posed by imprecise technical writing, saying that the first lesson to be learned is that "[...]the specification of a protocol should be sufficiently precise and explicit[...]". They also add that their experiences show that, even after formal proof of a specification's security, "[...]it is critical to keep auditing and testing actual implementations." This further emphasises the security deficit inherent in closed-source software, as well as in closed specifications. Green (2017) reports that "[o]ne of the problems with IEEE is that the standards are highly complex and get made via a closed-door process of private meetings. More importantly, even after the fact, theyre hard for ordinary security researchers to access." He concludes that "[t]his whole process is dumb and—in this specific case—probably just cost industry tens of millions of dollars."

Indeed, perhaps the IEEE could have avoided this whole embarassing affair if they had adhered to RFC 2119[2] rather than 6919[3], but Green further posits that part of the issue is the very nature of standard validation itself. "We need machine-assisted verification of protocols," he writes, "preferably tied to the actual source code that implements them." He claims that "[t]his would ensure that the protocol actually does what it says, and that implementers dont further screw it up, thus invalidating the security proof." However, he is sensitive that it is early days yet for this potential avenue of research.

However, by far the more immediate issue is how much of an impact this will have on the average user. As it stands now, many vendors either have released patches, or are expected to do so soon. Some devices will likely never be patched, but they are unlikely to be found in any security-critical environments—they tend to be predominantly cheaper, foreign-made IoT devices. Ultimately, the magnitude of the threat does not appear commensurate with the attention being paid to the discovery. Far from the warnings of "[...]cyber-crooks run[ning] riot across the globe[...]" seen in Hamill (2017), the most compelling discussions to come from this are around the complacency and practices that allowed a fundamental flaw such as this to go unnoticed for 14 years. Vitally, as Vanhoef is quick to stress, we do not now require a 'WPA3'—"[...]implementations can be patched in a backwards-compatible manner[...]". Thus, as Dr Steve Bagley puts it in Riley (2017), "it's a risk, it needs to be patched, it'll be patched, and then we can all go back to using Wi-Fi and browsing the Web."

[2] `https://tools.ietf.org/html/rfc2119`
[3] `https://tools.ietf.org/html/rfc6919`

References

Fluhrer, S., Mantin, I., Shamir, A. et al. (2001), Weaknesses in the key scheduling algorithm of rc4, *in* 'Selected areas in cryptography', Vol. 2259, Springer, pp. 1–24.

Fouque, P.-A., Martinet, G., Valette, F. & Zimmer, S. (2008), On the security of the ccm encryption mode and of a slight variant, *in* 'Applied Cryptography and Network Security', Springer, pp. 411–428.

Grand View Research (2016), *Wireless Gigabit (WiGig) Market Analysis By Type (802.11ac, 802.11ad), By Product (Consumer Electronics, Networking Devices), By Application (BFSI, Government, IT & Telecom, Retail, Healthcare), By End-Use (Large Enterprises, SMBs, Residential) And Segment Forecasts To 2024.*
URL: *http://www.grandviewresearch.com/industry-analysis/wireless-gigabit-wigig-market*

Green, M. (2017), 'Falling through the kracks'.
URL: *https://blog.cryptographyengineering.com/2017/10/16/falling-through-the-kracks/*

Hamill, J. (2017), 'Scary 'krack' vulnerability could let hackers crack into any wifi network on the planet'.
URL: *https://www.thesun.co.uk/tech/4695773/scary-krack-vulnerability-could-let-hackers-crack-into-any-wifi-network-on-the-planet/*

He, C., Sundararajan, M., Datta, A., Derek, A. & Mitchell, J. C. (2005), A modular correctness proof of ieee 802.11 i and tls, *in* 'Proceedings of the 12th ACM conference on Computer and communications security', ACM, pp. 2–15.

Leyden, J. (2017), 'Wpa2 krack attack smacks wi-fi security: Fundamental crypto crapto'.
URL: *https://www.theregister.co.uk/2017/10/16/wpa2_krack_attack_security_wifi_wireless/*

Riley, S. (2017), *Krack Attacks (WiFi WPA2 Vulnerability)*, Computerphile.

Stubblefield, A., Ioannidis, J., Rubin, A. D. et al. (2002), Using the fluhrer, mantin, and shamir attack to break wep., *in* 'NDSS'.

Thomson, I. (2017), 'Release the kracken patches: The good, the bad, and the ugly on this wpa2 wi-fi drama'.
URL: *https://www.theregister.co.uk/2017/10/17/kracken_patches/*

Vanhoef, M. (2017), 'Krack attacks: Breaking wpa2'.
URL: *https://www.krackattacks.com/*

Vanhoef, M. & Piessens, F. (2013), Practical verification of wpa-tkip vulnerabilities, *in* 'Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security', ACM, pp. 427–436.

Vanhoef, M. & Piessens, F. (2017), Key reinstallation attacks: Forcing nonce reuse in wpa2, *in* 'Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS)', ACM.